

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Group Art Unit No.: 2146
Rajiv Goel, et al. Examiner: Avellino, J. E.
Serial No.: 10/824,725 Confirmation No.: 6713
Filed on: April 14, 2004
For: DYNAMIC CHAIN CREATION AND
SEGMENTATION OF THE PACKET-
FORWARDING PLANE

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF (CORRECTED)

Sir:

This corrected Appeal Brief is submitted in support of the Notice of Appeal filed on December 23, 2008 and is timely filed in response to the Notification of Non-Compliant Appeal Brief mailed March 19, 2009.

I. REAL PARTY IN INTEREST

The real party in interest is Cisco Systems, Inc., which wholly owns the assignee Cisco Technology, Inc.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 25-45 and 47-57 are now pending in this application, were finally rejected, and are the subject of this appeal. Claims 1-24 and 46 have been cancelled.

IV. STATUS OF AMENDMENTS

Claim amendments for the sole purpose of canceling Claim 46 were filed with Appellants' original Appeal Brief. The Examiner has not reviewed and entered these claim amendments filed with the original Appeal Brief.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claim 25, 36, and 47. All references in this section refer to Appellant's application specification and drawings.

Independent Claim 25

Claim 25 is directed to a method of selectively creating chains for a virtual interface (*see, e.g.*, FIG. 2; FIG. 3), the method comprising the computer-implemented steps of:

determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface by determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface (*see, e.g.*, 306 of FIG. 3; page 21, line 22 – page 22, line 10, paragraph 65);

determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface by determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface (*see, e.g.*, 312 of FIG. 3; page 23, lines 1-12, paragraph 68);

in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new

encapsulation chain for the particular virtual interface (*see, e.g.*, 310 of FIG. 3; page 22, lines 18 – 23, paragraph 67); and

in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface (*see, e.g.*, 316 of FIG. 3; page 23, lines 19 – 23, paragraph 70).

Independent Claim 35

Claim 35 is directed to a method of selectively creating chains on a plurality of cards of a network router (*see, e.g.*, FIG. 2; FIG. 3), the method comprising the computer-implemented steps of:

for each particular card within the plurality of cards (204 of FIG. 2), performing the steps of:

for each particular virtual interface of the particular card (*see, e.g.*, page 20, lines 2 – 9, paragraphs 57 and 58), performing the steps of:

determining whether at least one physical port of the particular card is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface (*see, e.g.*, 306 of FIG. 3; page 21, line 22 – page 22, line 10, paragraph 65);

determining whether at least one physical port of the particular card is (a) configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface (*see, e.g.*, 312 of FIG. 3; page 23, lines 1-12, paragraph 68);

determining whether the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains

for the particular virtual interface (*see, e.g.*, 302 of FIG. 3; page 20, line 20 – page 21, line 9, paragraphs 61-63);

in response to determining that the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using no resources of the particular card to create an encapsulation chain and a decapsulation chain for the particular virtual interface (*see, e.g.*, 304 of FIG. 3; page 21, lines 7 – 21, paragraphs 63 and 64);

in response to determining that no physical port of the particular card is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface, then using no resources of the particular card to create an encapsulation chain for the particular virtual interface (*see, e.g.*, 308 of FIG. 3; page 22, lines 11 – 17, paragraph 66);

in response to determining that no physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, then using no resources of the particular card to create a decapsulation chain for the particular virtual interface (*see, e.g.*, 314 of FIG. 3; page 23, lines 13 – 18, paragraph 69);

in response to determining that at least one physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using resources of

the particular card to create an encapsulation chain for the particular virtual interface (*see, e.g.*, 310 of FIG. 3; page 22, lines 18 – 23, paragraph 67); and

in response to determining that at least one physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using resources of the particular card to create a decapsulation chain for the particular virtual interface (*see, e.g.*, 316 of FIG. 3; page 23, lines 19 – 23, paragraph 70).

Claim 36

Claim 36 is directed to a volatile or non-volatile computer-readable medium (*see, e.g.*, page 26, lines 4 – 11, paragraph 77) carrying one or more sequences of instructions for selectively creating chains for a virtual interface, which instructions, when executed by one or more processors, cause the one or more processors to carry out (*see, e.g.*, FIG. 2; FIG. 3; FIG. 4):

determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface by determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface (*see, e.g.*, 306 of FIG. 3; page 21, line 22 – page 22, line 10, paragraph 65);

determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface by determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that

would be processed by a decapsulation chain for the particular virtual interface (*see, e.g.*, 312 of FIG. 3; page 23, lines 1-12, paragraph 68);

in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new encapsulation chain for the particular virtual interface (*see, e.g.*, 310 of FIG. 3; page 22, lines 18 – 23, paragraph 67); and

in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface (*see, e.g.*, 316 of FIG. 3; page 23, lines 19 – 23, paragraph 70).

Claim 47

Claim 47 is directed to an apparatus for selectively creating chains for a virtual interface (*see, e.g.*, FIG. 2; FIG. 3; FIG. 4), comprising:

a network interface that is coupled to a data network for receiving one or more packet flows therefrom (*see, e.g.*, 414 and 419 of FIG. 4);

a processor (*see, e.g.*, 404 of FIG. 4);

one or more stored sequences of instructions (*see, e.g.*, 406 of FIG. 4; page 25, line 16 – page 26, line 3, paragraph 76) which, when executed by the processor, cause the processor to carry out:

determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface by determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can

send data packets toward a destination associated with the particular virtual interface (*see, e.g.*, 306 of FIG. 3; pages 21 and 22, paragraph 65);

determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface by determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface (*see, e.g.*, 312 of FIG. 3; page 23, lines 1-12, paragraph 68);

in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new encapsulation chain for the particular virtual interface (*see, e.g.*, 310 of FIG. 3; page 22, lines 18 – 23, paragraph 67); and

in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface (*see, e.g.*, 316 of FIG. 3; page 23, lines 19 – 23, paragraph 70).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether Claims 25-30, 34, 36-41, 45, 47-52, and 56 are unpatentable under 35 U.S.C. § 103 (a) over *Conta et al.*, U.S. Patent Pub. No. 2005/0086367 A1 (hereinafter *Conta*) in view of *Singh et al.*, U.S. Patent Pub. No. 2005/0108315 (hereinafter *Singh*).
2. Whether Claims 31-33, 35, 42-44, 53-55, and 57 are unpatentable under 35 U.S.C. § 103 (a) over *Conta* in view of *Singh*, further in view of *Tuniman et al.*, U.S. Patent No. 6,507,874 B1 (hereinafter *Tuniman*).

VII. ARGUMENTS

To establish the obviousness of the claimed subject matter, all the claim limitations must be taught or suggested by the prior art. *See In re Royka*, 490 F.2d 981, 985 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

A. Claims 25-30, 34, 36-41, 45, 47-52, and 56 Are Patentable over *Conta* in View of *Singh*.

Claim 25

Claim 25 recites, *inter alia*:

determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface;
wherein determining whether a new encapsulation chain should be created comprises:
determining whether at least one physical port of a particular card of the network element **(a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface;**
determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface;
wherein determining whether a new decapsulation chain should be created comprises:
determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface;
in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new encapsulation chain for the particular virtual interface; and
in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface.
(Emphasis added)

In contrast, *Conta* discloses a uniform method for data packets of multiple tunneling protocols in a switch or router (page 2, paragraph 32). In *Conta*, a tunnel is made up of a tunnel origination interface from a set of iL2 or iL3 streams and a tunnel termination

interface from a set of eL2 or eL3 streams (e.g., a tunnel may be made up of iL2n->T2->eL21 as illustrated in FIG. 1). At the tunnel origination point, as a data packet arrives from an interface a forwarding information base lookup that is based on the data packet to be tunneled yields a particular tunnel origination interface (e.g., 20, 22, 30, or 32 of FIG. 2) for the tunnel along which the data packet is to travel (pages 3 and 4, paragraph 54). Similarly, at the tunnel origination point, as a data packet is received by an interface, a forwarding information base lookup that is based on the received data packet yields a particular output tunnel interface (e.g., 50, 52, 60 or 62 of FIG. 3) for the tunnel along which the data packet is to travel (*Id.* at page 4, paragraph 62). *Conta* discloses an encapsulation engine (21 of FIG. 2) and a decapsulation engine (61 of FIG. 3). However, *Conta* is devoid of a description of when these engines are created. Rather, *Conta* only describes that these two engines can be invoked to prepend or drop headers when data packets are processed (e.g., page 4, paragraphs 55 and 63).

Singh is cited by Examiner to show only that a network element can have multiple cards.

Conta and *Singh*, taken individually or in combination, fail to disclose each and every feature of Claim 25 for at least the following reasons.

1. Both *Conta* and *Singh* Fails to Disclose the Determination Steps of Claim 25 as Related to the New Decapsulation Chain.

Analogizing the encapsulation engine and the decapsulation engine of *Conta* to the new encapsulation chain and the new decapsulation chain of Claim 25, respectively, the Examiner at pages 3, line 18 – page 4, line 5 of the Final Office Action mailed August 29, 2008 (hereinafter “the Office Action”) argues that *Conta* at paragraphs 33, 58, and 81 and FIGs. 2-4 discloses “determining whether a new decapsulation chain should be created

comprises: determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface”, as featured in Claim 25. This is clear error.

The cited excerpts in *Conta* only describe how a received packet is processed in *Conta*. At best, these excerpts describe decapsulating the packet using an already created decapsulation engine. But *Conta* is devoid of any description of determining whether a new decapsulation chain should be created, as recited in Claim 25. In fact, the decapsulation engine of *Conta* is mentioned just three times in paragraphs 62-64 of *Conta*. None of these paragraphs describes how to determine whether a new decapsulation engine should be created, much less determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, as claimed. Claim 25, on the other hand, features a determination of whether to create a decapsulation chain in response to a determination of port configuration (i.e., whether a port is configured to receive certain packets), while *Conta* describes neither a port level determination nor a determination of whether to create a decapsulation chain.

As *Conta* only describes that a decapsulation engine is invoked to process a packet that is received from a tunnel, by the time the decapsulation engine is invoked to process the packet for transmission, the packet already would be received at a physical port of the *Conta* system. *Conta* has no sensible reason or need to determine at that point whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface. The claimed features are simply missing in *Conta*.

Since *Singh* also fails to disclose the claimed features, *Conta* and *Singh* together fail to describe either explicitly or inherently “determining whether a new decapsulation chain should be created comprises: determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface” as claimed.

2. *Conta* and *Singh* Together Fails to Disclose the Determination Steps of Claim 25 as Related to the New Encapsulation Chain.

Page 3, lines 7 – 17 of the Office Action also argues that *Conta* at paragraphs 33, 58, and 81 and FIGs. 2-4 discloses “determining whether a new encapsulation chain should be created comprises: determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface”, as featured in Claim 25. The Examiner’s argument is clear error.

These cited excerpts in *Conta* only describe how a packet is processed for transmission in *Conta*. At best, these excerpts describe encapsulating the packet using an already created encapsulation engine. *Conta* is devoid of any description of the determining steps, as recited in Claim 25, as related to the new encapsulation chain.

The encapsulation engine of *Conta* is only mentioned four times in *Conta* at paragraphs 55, 56, 58, and 59. None of these paragraphs describes determining whether a new encapsulation engine should be created, much less determining whether a new encapsulation chain should be created that comprises determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual

interface and (b) can send data packets toward a destination associated with the particular virtual interface. The claimed features are simply missing in *Conta*.

Since *Singh* also fails to disclose the claimed features, *Conta* and *Singh* together fail to describe either explicitly or inherently “determining whether a new encapsulation chain should be created comprises: determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface” as claimed.

3. Examiner’s Argument Lacks Requisite Factual Support and Is Erroneous.

The Examiner at page 3, lines 13 – 17 and page 4, lines 1 – 5 of the Office Action argues that “a tunnel endpoint is inherently created in order to establish communication via a tunnel, and based on the particular endpoint type will determine whether an encapsulation engine should be created, and will create one if the tunnel interface is a transmitting interface” and that “a tunnel endpoint is inherently created in order to establish communication via a tunnel, and based on the particular endpoint type, such as sending or receiving, will determine whether a decapsulation engine should be created, and will create one if the tunnel interface is a receiving interface.” This inherency-based argument is unsupported and incorrect.

Even if a tunnel endpoint is inherently created, as alleged by Examiner, *Conta* does not explicitly or inherently disclose the determinations of creating encapsulation and decapsulation chains as featured in Claim 25. Indeed, in *Conta*, encapsulation and decapsulation engines might be created when the network element is installed. They might be created without determining whether a port is configured to receive certain types of data

packets. These engines of *Conta* might be implemented in a monolithic package with the operating system of a network element, and any part of the engines might be installed at the same time as other parts of the engines, without any (intelligent) determination of whether any port is configured for certain data packets. There are simply many reasonable explanations why the engines of *Conta* may not inherently involve determining whether the engines should be created and in particular, determining whether the engines should be created in response to determining whether a port is configured to send or receive certain types of packets. There is no reason that *Conta* necessarily use the claimed approach.

Simply put, Examiner's inherency-based argument lacks factual support and is incorrect.

Singh is likewise devoid of any mention of the features of Claim 25 that are missing from *Conta*.

Based on at least the reasons given above, Claim 25 is patentable over *Conta* in view of *Singh*. Reconsideration and removal of the rejection to Claim 25 is respectfully requested.

Claims 36, 46, and 47

Claims 36, 46, and 47 each recite similar features as those discussed above with respect to Claim 25. Therefore, Claims 36, 46, and 47 are patentable for at least the same reasons discussed above as to Claim 25.

Claims 26-30, 34, 37-41, 45, and 48-52

Claims 26-30, 34, 37-41, 45, and 48-52 depend from, and hence, incorporate all of the limitations of Claim 25, 36, 46, or 47. Applicant submits that Claims 26-30, 34, 37-41, 45, and 48-52 are patentable for at least the reasons given above in connection with Claim 25, 36, 46, or 47.

B. Claims 31-33, 35, 42-44, 53-55, and 57 Are Patentable over *Conta* in View of *Singh* and in Further View of *Tuniman*.

Claim 57

Claim 57 recites similar features as those discussed above with respect to Claim 25. *Tuniman* fails to disclose those features of Claim 25 that are missing in *Conta* and *Singh*. *Tuniman* is not even related to encapsulation or decapsulation, let alone creating a chain for encapsulation or decapsulation. Therefore, Claim 57 is patentable for at least the same reasons discussed above as to Claim 25.

Claims 31-33, 35, 42-44, and 53-55

Claims 31-33, 35, 42-44, 53-55 depend from, and hence, incorporate all of the limitations of Claim 25, 36, 46, or 47. For the reasons previously noted in connection to Claim 57, *Tuniman* fails to disclose those features of Claim 25, 36, 46, or 47 that are missing in *Conta* and *Singh*. Applicant submits that Claims 31-33, 35, 42-44, 53-55 are patentable for at least the reasons given above in connection with Claim 25, 36, 46, or 47.

VIII. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that (1) the rejection of Claims 25-30, 34, 36-41, 45-52, and 56 under 35 U.S.C. § 103 (a) over *Conta* in view of *Singh* and (2) the rejection of Claims 31-33, 35, 42-44, 53-55, and 57 under 35 U.S.C. § 103 (a) over *Conta* in view of *Singh*, further in view of *Tuniman*, lacks the requisite factual and legal bases. Appellants therefore respectfully request that the Honorable Board reverse the rejections of Claims 25-45 and 47-57.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER
LLP

/ZhichongGu#56543/

Zhichong Gu

Reg. No. 56,543

Date: April 8, 2009

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1236
Fax: (408) 414-1076

IX. CLAIMS APPENDIX

1-24. (Canceled)

25. (Previously Presented) A method of selectively creating chains for a virtual interface, the method comprising the computer-implemented steps of:
determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface by determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface;
determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface by determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface;
in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new encapsulation chain for the particular virtual interface; and
in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface.
26. (Previously Presented) A method as recited in Claim 25, further comprising:
in response to determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface, avoiding creating, on the network element, a new encapsulation chain for the particular virtual interface.
27. (Previously Presented) A method as recited in Claim 25, further comprising:
in response to determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface, avoiding creating, on

the network element, a new decapsulation chain for the particular virtual interface.

28. (Previously Presented) A method as recited in Claim 25, further comprising:
in response to determining that a new encapsulation chain should not be created, on
the network element, for the particular virtual interface, avoiding creating, on
the network element, a new encapsulation chain for the particular virtual
interface; and
in response to determining that a new decapsulation chain should not be created, on
the network element, for the particular virtual interface, avoiding creating, on
the network element, a new decapsulation chain for the particular virtual
interface.
29. (Previously Presented) A method as recited in Claim 25, wherein the step of
determining whether a new encapsulation chain should be created further comprises
the steps of:
in response to determining that no physical port of the particular card (a) is
configured to send data packets of a type that would be produced by an
encapsulation chain for the particular virtual interface and (b) can send data
packets toward the destination associated with the particular virtual interface,
then determining that a new encapsulation chain should not be created, on the
network element, for the particular virtual interface.
30. (Previously Presented) A method as recited in Claim 25, wherein the step of
determining whether a new decapsulation chain should be created further comprises
the steps of:
in response to determining that no physical port of the particular card is configured to
receive data packets of a type that would be processed by a decapsulation
chain for the particular virtual interface, then determining that a new
decapsulation chain should not be created, on the network element, for the
particular virtual interface.

31. (Previously Presented) A method as recited in Claim 25, wherein the steps of determining whether a new encapsulation chain should be created and determining whether a new decapsulation chain should be created comprise the steps of:
determining whether a plurality of cards of the network element includes a
specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and
in response to determining that the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface and determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.
32. (Previously Presented) A method as recited in Claim 25, wherein the step of determining whether a new encapsulation chain should be created further comprises the steps of:
determining whether a plurality of cards of the network element includes a
specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and
in response to determining that at least one physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface.

33. (Previously Presented) A method as recited in Claim 25, wherein the step of determining whether a new decapsulation chain should be created further comprises the steps of:
determining whether a plurality of cards of the network element includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and
in response to determining that at least one physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface.
34. (Previously Presented) A method as recited in Claim 25, wherein determining whether a new encapsulation chain should be created and determining whether a new decapsulation chain should be created are based on user input.
35. (Previously Presented) A method of selectively creating chains on a plurality of cards of a network router, the method comprising the computer-implemented steps of:
for each particular card within the plurality of cards, performing the steps of:
for each particular virtual interface of the particular card, performing the steps of:
determining whether at least one physical port of the particular card is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface;
determining whether at least one physical port of the particular card is (a) configured to receive data packets of a type that would be

processed by a decapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface;

determining whether the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface;

in response to determining that the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using no resources of the particular card to create an encapsulation chain and a decapsulation chain for the particular virtual interface;

in response to determining that no physical port of the particular card is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface, then using no resources of the particular card to create an encapsulation chain for the particular virtual interface;

in response to determining that no physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, then using no resources of the particular card to create a decapsulation chain for the particular virtual interface;

in response to determining that at least one physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing

that would be performed by one or more chains for the particular virtual interface, then using resources of the particular card to create an encapsulation chain for the particular virtual interface; and

in response to determining that at least one physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using resources of the particular card to create a decapsulation chain for the particular virtual interface.

36. (Previously Presented) A volatile or non-volatile computer-readable medium carrying one or more sequences of instructions for selectively creating chains for a virtual interface, which instructions, when executed by one or more processors, cause the one or more processors to carry out:

determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface by determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface;

determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface by determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface;

in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new encapsulation chain for the particular virtual interface; and

in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface.

37. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out avoiding creating, on the network element, a new encapsulation chain for the particular virtual interface, in response to determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface.
38. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out avoiding creating, on the network element, a new decapsulation chain for the particular virtual interface, in response to determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.
39. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:
avoiding creating, on the network element, a new encapsulation chain for the particular virtual interface, in response to determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface; and
avoiding creating, on the network element, a new decapsulation chain for the particular virtual interface, in response to determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.
40. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further

comprise instructions which, when executed, cause the one or more processor to carry out:

in response to determining that no physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, then determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface.

41. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:

in response to determining that no physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, then determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.

42. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:

determining whether a plurality of cards of the network element includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and

in response to determining that the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface and determining that a

new decapsulation chain should not be created, on the network element, for the particular virtual interface.

43. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:

determining whether a plurality of cards of the network element includes a

specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and

in response to determining that at least one physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface.

44. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein the step of the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:

determining whether a plurality of cards of the network element includes a

specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and

in response to determining that at least one physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, and the plurality of

cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface.

45. (Previously Presented) A volatile or non-volatile computer-readable medium as recited in Claim 36, wherein determining whether a new encapsulation chain should be created and determining whether a new decapsulation chain should be created are based on user input.
46. (Canceled)
47. (Previously Presented) An apparatus for selectively creating chains for a virtual interface, comprising:
a network interface that is coupled to a data network for receiving one or more packet flows therefrom;
a processor;
one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out:
determining whether a new encapsulation chain should be created, on a network element, for a particular virtual interface by determining whether at least one physical port of a particular card of the network element (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface;
determining whether a new decapsulation chain should be created, on the network element, for the particular virtual interface by determining whether at least one physical port of a particular card of the network element is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface;

- in response to determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new encapsulation chain for the particular virtual interface; and in response to determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface, creating, on the network element, a new decapsulation chain for the particular virtual interface.
48. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out avoiding creating, on the network element, a new encapsulation chain for the particular virtual interface, in response to determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface.
49. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out avoiding creating, on the network element, a new decapsulation chain for the particular virtual interface, in response to determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.
50. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:
avoiding creating, on the network element, a new encapsulation chain for the particular virtual interface, in response to determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface; and
avoiding creating, on the network element, a new decapsulation chain for the particular virtual interface, in response to determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.

51. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:
in response to determining that no physical port of the particular card (a) is
configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, then determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface.
52. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:
in response to determining that no physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, then determining that a new decapsulation chain should not be created, on the network element, for the particular virtual interface.
53. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:
determining whether a plurality of cards of the network element includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and
in response to determining that the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new encapsulation chain should not be created, on the network element, for the particular virtual interface and determining that a

new decapsulation chain should not be created, on the network element, for the particular virtual interface.

54. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:

determining whether a plurality of cards of the network element includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and

in response to determining that at least one physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then determining that a new encapsulation chain should be created, on the network element, for the particular virtual interface.

55. (Previously Presented) An apparatus as recited in Claim 47, wherein the one or more sequences of instructions further comprise instructions which, when executed, cause the one or more processor to carry out:

determining whether a plurality of cards of the network element includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface; and

in response to determining that at least one physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for

the particular virtual interface, then determining that a new decapsulation chain should be created, on the network element, for the particular virtual interface.

56. (Previously Presented) An apparatus as recited in Claim 47, wherein determining whether a new encapsulation chain should be created and determining whether a new decapsulation chain should be created are based on user input.
57. (Previously Presented) A method of selectively creating chains on a plurality of cards of a network router, the method comprising the computer-implemented steps of: for each particular card within the plurality of cards, performing the steps of:
- for each particular virtual interface of the particular card, performing the steps of:
 - determining whether at least one physical port of the particular card is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface;
 - determining whether at least one physical port of the particular card is (a) configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface and (b) can send data packets toward a destination associated with the particular virtual interface;
 - determining whether the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface;
 - in response to determining that the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using no resources of the particular card to create an encapsulation chain and a decapsulation chain for the particular virtual interface;
 - in response to determining that no physical port of the particular card is configured to send data packets of a type that would be produced

- by an encapsulation chain for the particular virtual interface, then using no resources of the particular card to create an encapsulation chain for the particular virtual interface;
- in response to determining that no physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, then using no resources of the particular card to create a decapsulation chain for the particular virtual interface;
- in response to determining that at least one physical port of the particular card (a) is configured to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b) can send data packets toward the destination associated with the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using resources of the particular card to create an encapsulation chain for the particular virtual interface; and
- in response to determining that at least one physical port of the particular card is configured to receive data packets of a type that would be processed by a decapsulation chain for the particular virtual interface, and the plurality of cards does not include any specialized card that is designed to perform a type of data packet processing that would be performed by one or more chains for the particular virtual interface, then using resources of the particular card to create a decapsulation chain for the particular virtual interface.

X. EVIDENCE APPENDIX

None

XI. RELATED PROCEEDINGS INDEX

None